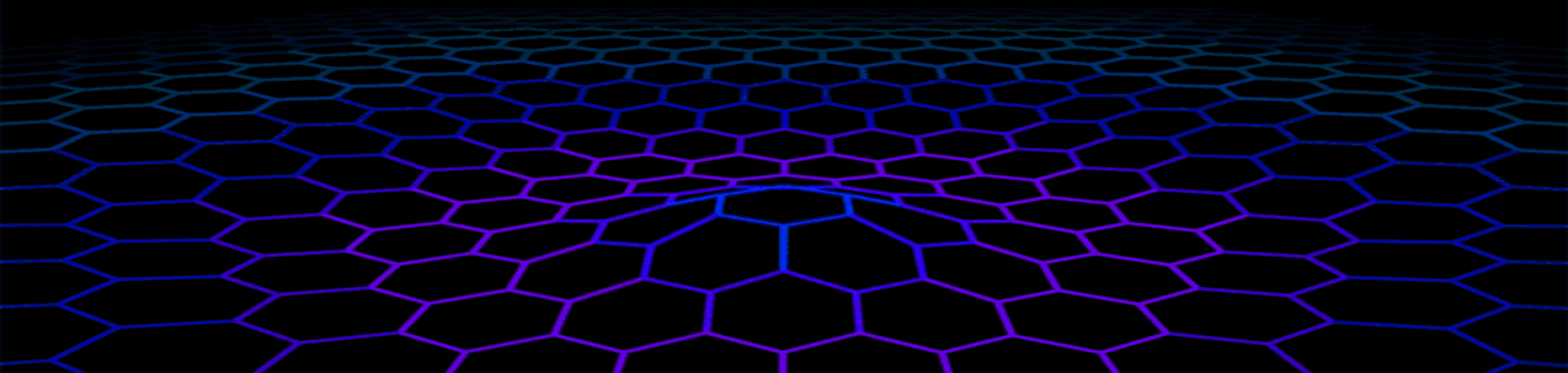


Sistemas Operacionais



Te acalma que nem começou.
Agora que vai ficar bom.

Computadores hipotéticos



Computadores hipotéticos

Programa

P	End.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

BP: 255 [0]: 0

Neander

Arquivo Editar Visualizar Executar ?

AC: 888 PC: 888

Execução: 888888

Acessos: 888888

Instruções: 888888

Instrução: Reg.Instrução: 0 Mnemônico: NOP

Ok.

Mnemônicos

Mnemônico	End.	Dado
NOP	00	
STA	16 end	
LDA	32 end	
ADD	48 end	
OR	64 end	
AND	80 end	
NOT	96	
JMP	128 end	
JN	144 end	
JZ	160 end	
HLT	240	

[128]: 0

Programa

P	End.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

BP: 255 [3]: 0

Ahmes

Arquivo Editar Visualizar Executar ?

AC: 888 PC: 888

Execução: 888888

Acessos: 888888

Instr.: 888888

Instrução: R I: 0 Mnem: NOP

Ok.

Mnemônicos

Mnemônico	End.	Dado
NOP	00	
STA	16 end	
LDA	32 end	
ADD	48 end	
OR	64 end	
AND	80 end	
NOT	96	
SUB	112 end	
JMP	128 end	
JN	144 end	
JP	148 end	
JV	152 end	
JNV	156 end	
JZ	160 end	
JNZ	164 end	
JC	176 end	
JNC	180 end	
JB	184 end	
JNB	188 end	
SHR	224	
SHL	225	
ROR	226	
ROL	227	
HLT	240	

[128]: 0

Programa

P	End.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

BP: 255 [0]: 0

Ramses v1.3

Arquivo Editar Visualizar Executar ?

RA: 888 RB: 888 RX: 888 PC: 888

Execução: 888888

Acessos: 888888

Instr.: 888888

Instrução: R I: 0 Mnem: NOP

Ok.

Códigos das instruções

Mnemônico	End.	Dado
NOP	0	
STR	16 r end	
LDR	32 r end	
ADD	48 r end	
OR	64 r end	
AND	80 r end	
NOT	96 r	
SUB	112 r end	
JMP	128 end	
JN	144 end	
JZ	160 end	
JC	176 end	
JSR	192 end	
NEG	208 r	
SHR	224 r	
HLT	240	

Modo: 0: Dir: n 1: Ind: n,l 2: Ind: #n 3: Idx: n,X

Registrador: 0: A 2: X 1: B 3: ?

[128]: 0

Programa

P	Ender.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

BP: 65535 [0]: 0

Cesar 16i

Arquivo Editar Visualizar Executar ?

R0: 88888 R1: 88888 R2: 88888

R3: 88888 R4: 88888 R5: 88888

R6: (SP) 88888 IS R7: (PC) 88888

Execução: 888888

Acessos: 888888

Instr.: 888888

Instrução: R I: 0 Mnem: NOP

Pronto

Dados

Ender.	Dado
1024	0
1025	0
1026	0
1027	0
1028	0
1029	0
1030	0
1031	0
1032	0
1033	0
1034	0
1035	0
1036	0
1037	0
1038	0
1039	0

[1024]: 0

Computador NEANDER

The image shows a screenshot of the Neander computer simulator interface, which consists of three main windows: Programa, Neander, and Dados.

Programa Window: Displays a list of instructions in a table with columns P, End., Dado, and Mnemônico.

P	End.	Dado	Mnemônico
→	0	0	NOP
	1	0	NOP
	2	0	NOP
	3	0	NOP
	4	0	NOP
	5	0	NOP
	6	0	NOP
	7	0	NOP
	8	0	NOP
	9	0	NOP
	10	0	NOP
	11	0	NOP
	12	0	NOP
	13	0	NOP
	14	0	NOP
	15	0	NOP

At the bottom of the Programa window, there are input fields for **BP:** 255 and **[0]:** 0.

Neander Window: The central window titled "Neander" contains a menu bar (Arquivo, Editar, Visualizar, Executar, ?) and a status bar (Ok.). It features several digital displays and controls:

- AC:** Accumulator register display showing 000.
- PC:** Program Counter display showing 000.
- Execução:** Execution status area with **Acessos:** 00000 and **Instruções:** 00000.
- Instrução:** Instruction display showing 0.
- Reg.Instrução:** Register instruction display showing 0.
- Mnemônico:** Mnemonic display showing NOP.
- Controls:** Buttons for 0..9, 0..F, and icons for program execution (play, stop, reset).
- Flags:** N (Negative) and Z (Zero) status indicators.

Mnemônicos Window: A small window titled "Mnemônicos" listing the instruction set:

Mnemonic	Address	End
NOP	00	
STA	16	end
LDA	32	end
ADD	48	end
OR	64	end
AND	80	end
NOT	96	
JMP	128	end
JN	144	end
JZ	160	end
HLT	240	

Dados Window: A window titled "Dados" showing a memory dump table with columns End. and Dado.

End.	Dado
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0

At the bottom of the Dados window, there is an input field for **[128]:** 0.

Computador AHMES

Programa

P	End.	Dado	Mnemônico
→	0	0	NOP
	1	0	NOP
	2	0	NOP
	3	0	NOP
	4	0	NOP
	5	0	NOP
	6	0	NOP
	7	0	NOP
	8	0	NOP
	9	0	NOP
	10	0	NOP
	11	0	NOP
	12	0	NOP
	13	0	NOP
	14	0	NOP
	15	0	NOP

BP: 255 [3]: 0

Ahmes

Arquivo Editar Visualizar Executar ?

AC: 000

PC: 000

N Z V C B

Execução:

Acessos: 00000

Instr.: 00000

Instrução:

R I: 0

Mnem: NOP

0..9 0..F

Ok.

Mnemônicos

NOP	00	JMP	128 end	SHR	224
STA	16 end	JN	144 end	SHL	225
LDA	32 end	JP	148 end	ROR	226
ADD	48 end	JV	152 end	ROL	227
OR	64 end	JNV	156 end	HLT	240
AND	80 end	JZ	160 end		
NOT	96	JNZ	164 end		
SUB	112 end	JC	176 end		
		JNC	180 end		
		JB	184 end		
		JNB	188 end		

Dados

End.	Dado
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0

[128]: 0

Computador RAMSES

The image displays the RAMSES computer simulator interface, which includes three main windows: 'Programa', 'Ramses v1.3', and 'Dados'.

Programa Window: A table showing the program memory. The first instruction at address 0 is NOP. The base pointer (BP) is 255 and the accumulator (A) is 0.

P	End.	Dado	Mnemônico
→	0	0	NOP
	1	0	NOP
	2	0	NOP
	3	0	NOP
	4	0	NOP
	5	0	NOP
	6	0	NOP
	7	0	NOP
	8	0	NOP
	9	0	NOP
	10	0	NOP
	11	0	NOP
	12	0	NOP
	13	0	NOP
	14	0	NOP
	15	0	NOP

BP: 255 [0]: 0

Ramses v1.3 Window: The central window showing the execution status. It includes registers RA, RB, RX, and PC, all displaying 000. Status flags N, Z, and C are shown. The execution section displays the current instruction as NOP at address 0. The menu bar includes Arquivo, Editar, Visualizar, Executar, and ?.

RA: 000 RB: 000 RX: 000 PC: 000

N Z C

Execução:

Acessos: 00000 Instrução: RI: 0 Mnem: NOP

Instr.: 00000

0..9 0..F [Icons]

Ok.

Códigos das instruções Window: A table listing the instruction codes and their modes. The instructions are NOP, STR, LDR, ADD, OR, AND, NOT, SUB, JMP, JN, JZ, JC, JSR, NEG, SHR, and HLT. The modes are 0: Dir, 1: Ind, 2: Ind, 3: Ind, and 4: Ind.

Instruction	Code	Mode
NOP	0	0: Dir
STR	16	1: Ind
LDR	32	2: Ind
ADD	48	3: Ind
OR	64	4: Ind
AND	80	0: Dir
NOT	96	1: Ind
SUB	112	2: Ind
JMP	128	3: Ind
JN	144	4: Ind
JZ	160	0: Dir
JC	176	1: Ind
JSR	192	2: Ind
NEG	208	3: Ind
SHR	224	4: Ind
HLT	240	0: Dir

Modo:

0: Dir 1: Ind 2: Ind 3: Ind 4: Ind

Registrador:

0: A 1: B 2: X 3: ?

Dados Window: A table showing the data memory. The first data item at address 128 is 0. The base pointer (BP) is 255 and the accumulator (A) is 0.

End.	Dado
128	0
129	0
130	0
131	0
132	0
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	0
141	0
142	0
143	0

[128]: 0

Computador CESAR

The image shows the Cesar 16i emulator interface, which is divided into three main panels: Programa, Cesar 16i, and Dados.

Programa Panel: Displays a list of memory addresses and instructions. The first instruction at address 0 is NOP. The address 65535 is shown in the BP register, and the value 0 is shown in the [0] register.

P	Ender.	Dado	Mnemônico
0	0	0	NOP
1	0	0	NOP
2	0	0	NOP
3	0	0	NOP
4	0	0	NOP
5	0	0	NOP
6	0	0	NOP
7	0	0	NOP
8	0	0	NOP
9	0	0	NOP
10	0	0	NOP
11	0	0	NOP
12	0	0	NOP
13	0	0	NOP
14	0	0	NOP
15	0	0	NOP

Cesar 16i Panel: Displays the state of the processor registers and execution status. The registers R0 through R7 are shown, along with the instruction register (Instr.) and the instruction being executed (NOP). The status flags N, Z, V, and C are also displayed.

Dados Panel: Displays the state of the data memory. The address 1024 is shown in the [1024] register, and the value 0 is shown in the Dado column.

Ender.	Dado
1024	0
1025	0
1026	0
1027	0
1028	0
1029	0
1030	0
1031	0
1032	0
1033	0
1034	0
1035	0
1036	0
1037	0
1038	0
1039	0

The bottom of the interface shows a blue bar with a grid pattern.

Principais Características (Neander X Ahmes)

NEANDER

Largura de dados e de endereços de 8 bits

Modo de endereçamento: direto

1 registrador acumulador de 8 bits

1 apontador de programa de 8 bits

1 registrador de estado com **2** códigos de condição: negativo e zero

AHMES

Largura de dados e de endereços de 8 bits

Modo de endereçamento: direto

1 registrador acumulador de 8 bits

1 apontador de programa de 8 bits

1 registrador de estado com **5** códigos de condição: negativo, zero, **carry**, **borrow** e **overflow**

Conjunto de instruções (Neander X Ahmes)

Neander

Mnemônicos

NOP	00	ADD	48 end	JMP	128 end
STA	16 end	OR	64 end	JN	144 end
LDA	32 end	AND	80 end	JZ	160 end
		NOT	96	HLT	240

Ahmes

Mnemônicos

NOP	00		JMP	128	end	SHR	224
STA	16	end	JN	144	end	SHL	225
LDA	32	end	JP	148	end	ROR	226
ADD	48	end	JV	152	end	ROL	227
OR	64	end	JNV	156	end	HLT	240
AND	80	end	JZ	160	end		
NOT	96		JNZ	164	end		
SUB	112	end	JC	176	end		
			JNC	180	end		
			JB	184	end		
			JNB	188	end		

Conjunto de instruções do NEANDER

Código	Mnemônico	Operação
00	NOP	Nenhuma operação
16	STA end	Armazena acumulador no endereço “end” da memória
32	LDA end	Carrega o acumulador com o conteúdo do endereço “end” da memória
48	ADD end	Soma o conteúdo do endereço “end” da memória ao acumulador
64	OR end	Efetua operação lógica “OU” do conteúdo do endereço “end” da memória ao acumulador
80	AND end	Efetua operação lógica “E” do conteúdo do endereço “end” da memória ao acumulador
96	NOT	Inverte todos os bits do acumulador
128	JMP end	Desvio incondicional para o endereço “end” da memória
144	JN end	Desvio condicional, se “N=1”, para o endereço “end” da memória (Jump on N egative)
160	JZ end	Desvio condicional, se “Z=1”, para o endereço “end” da memória (Jump on Z ero)
240	HLT	Para o ciclo de busca-decodificação-execução

Conjunto de instruções do NEANDER

CÓDIGO	INSTRUÇÃO	COMENTÁRIO	AÇÕES
0000	NOP	Nenhuma Operação	Nenhuma operação
0001	STA end	Armazena acumulador (<i>store</i>)	$\text{MEM}(\text{end}) \leftarrow \text{AC}$
0010	LDA end	Carrega acumulador (<i>load</i>)	$\text{AC} \leftarrow \text{MEM}(\text{end})$
0011	ADD end	Soma	$\text{AC} \leftarrow \text{MEM}(\text{end}) + \text{AC}$
0100	OR end	OU lógico	$\text{AC} \leftarrow \text{MEM}(\text{end}) \text{ OR } \text{AC}$
0101	AND end	AND lógico	$\text{AC} \leftarrow \text{MEM}(\text{end}) \text{ AND } \text{AC}$
0110	NOT	Inverte acumulador (complementa)	$\text{AC} \leftarrow \text{NOT } \text{AC}$
1000	JMP end	Desvio incondicional (<i>jump</i>)	$\text{PC} \leftarrow \text{end}$
1001	JN end	Desvio condicional (<i>jump on negative</i>)	IF N=1 THEN $\text{PC} \leftarrow \text{end}$
1010	JZ end	Desvio condicional (<i>jump on zero</i>)	IF Z=1 THEN $\text{PC} \leftarrow \text{end}$
1111	HLT	Término de execução (<i>halt</i>)	pára processamento

Conjunto de instruções do AHMES

Código	Mnemônico	Operação
00	NOP	Nenhuma operação
16	STA end	Armazena acumulador no endereço “end” da memória
32	LDA end	Carrega o acumulador com o conteúdo do endereço “end” da memória
48	ADD end	Soma o conteúdo do endereço “end” da memória ao acumulador
64	OR end	Efetua operação lógica “OU” do conteúdo do endereço “end” da memória ao acumulador
80	AND end	Efetua operação lógica “E” do conteúdo do endereço “end” da memória ao acumulador
96	NOT	Inverte todos os bits do acumulador
112	SUB end	Subtrai o conteúdo do endereço “end” da memória do acumulador
128	JMP end	Desvio incondicional para o endereço “end” da memória
144	JN end	Desvio condicional, se “N=1”, para o endereço “end” da memória (Jump on N egative)
148	JP end	Desvio condicional, se “N=0”, para o endereço “end” da memória (Jump on P ositive)
152	JV end	Desvio condicional, se “V=1”, para o endereço “end” da memória (Jump on oV erflow)
156	JNV end	Desvio condicional, se “V=0”, para o endereço “end” da memória (Jump on Not oV erflow)
160	JZ end	Desvio condicional, se “Z=1”, para o endereço “end” da memória (Jump on Z ero)
164	JNZ end	Desvio condicional, se “Z=0”, para o endereço “end” da memória (Jump on Not Z ero)
176	JC end	Desvio condicional, se “C=1”, para o endereço “end” da memória (Jump on C arry On)
180	JNC end	Desvio condicional, se “C=0”, para o endereço “end” da memória (Jump on Not C arry On)
184	JB end	Desvio condicional, se “B=1”, para o endereço “end” da memória (Jump on B orrow)
188	JNB end	Desvio condicional, se “B=0”, para o endereço “end” da memória (Jump on Not B orrow)
224	SHR	Desloca o acumulador para a direita; o bit 7 do acumulador recebe 0; o Carry recebe o bit 0 do acumulador
225	SHL	Desloca o acumulador para a esquerda, o bit 0 do acumulador recebe 0; o Carry recebe o bit 7 do acumulador
226	ROR	Gira o acumulador para a direita; o bit 7 do acumulador recebe o Carry; o Carry recebe o bit 0 do acumulador
227	ROL	Gira o acumulador para a esquerda, o bit 0 do acumulador recebe o Carry; o Carry recebe o bit 7 do acumulador
240	HLT	Para o ciclo de busca-decodificação-execução

Computadores hipotéticos

Neander, Ahmes, Ramses e Cesar são simuladores de computadores hipotéticos

Prof. Raul F. Weber (UFRGS)



Downloads

Neander

<http://www.inf.ufrgs.br/arq/wiki/lib/exe/fetch.php?media=wneander.zip>

Ahmes

<http://www.inf.ufrgs.br/arq/wiki/lib/exe/fetch.php?media=wahmes.zip>

Ramses

<http://www.inf.ufrgs.br/arq/wiki/lib/exe/fetch.php?media=wramses.zip>

Cesar

<http://www.inf.ufrgs.br/arq/wiki/lib/exe/fetch.php?media=wcesar16.1.3.5.2.exe.zip>

Mussum => Neander + Ahmes + Ramses + Cesar

<http://177.101.203.139/gladimir/SO/>

Web

Web – Neander

<https://www.inf.ufrgs.br/~vbuaraujo/sw/neander/neander.html>

Web – Ahmes

<https://www.inf.ufrgs.br/~vbuaraujo/sw/neander/ahmes.html>

Soma de dois (inteiros maiores do que zero)

Programa (*)

P	End.	Dado	Mnemônico
	0	33	LDA 131
1	131		
2	48		ADD 132
3	132		
4	16		STA 140
5	140		
6	0		NOP
7	0		NOP
8	0		NOP
9	0		NOP
10	0		NOP
11	0		NOP
12	0		NOP
13	0		NOP
14	0		NOP
15	0		NOP

BP: 255 [0]: 33

Neander

Arquivo Editar Visualizar Executar ?

AC: 000
N

PC: 000
Z

Execução:
Acessos: 00000
Instruções: 00000
Instrução:
Reg.Instrução: 0
Mnemônico: NOP

Ok.

Mnemônicos

NOP	00	ADD	48 end	JMP	128 end
STA	16 end	OR	64 end	JN	144 end
LDA	32 end	AND	80 end	JZ	160 end
		NOT	96	HLT	240

Dados

End.	Dado
128	0
129	0
130	0
131	13
132	20
133	0
134	0
135	0
136	0
137	0
138	0
139	0
140	33
141	0
142	0
143	0

[131]: 13

Multiplicação entre dois números (inteiros maiores do que zero)

Programa (*)

P	End.	Dado	Mnemônico
→	0	32	LDA 129
	1	129	
	2	16	STA 135
	3	135	
	4	32	LDA 130
	5	130	
	6	48	ADD 128
	7	128	
	8	16	STA 130
	9	130	
	10	32	LDA 135
	11	135	
	12	112	SUB 136
	13	136	
	14	16	STA 135
	15	135	
	16	160	JZ 30
	17	30	
	18	128	JMP 4
	19	4	
	20	0	NOP
	21	0	NOP
	22	0	NOP

Ahmes

Arquivo Editar Visualizar Executar ?

AC:

PC:

N

Z

V

C

B

Execução:

Acessos:

Instr.:

R I: 160 30

Mnem: JZ 30

0..9 0..F

Ok.

Mnemônicos

NOP	0	JMP	128 end	SHR	224
STA	16 end	JN	144 end	SHL	225
LDA	32 end	JP	148 end	ROR	226
ADD	48 end	JV	152 end	ROL	227
OR	64 end	JNV	156 end	HLT	240
AND	80 end	JZ	160 end		
NOT	96	JNZ	164 end		
SUB	112 end	JC	176 end		
		JNC	180 end		
		JB	184 end		
		JNB	188 end		

Dados

End.	Dado
128	8
129	5
130	0
131	0
132	0
133	0
134	0
135	0
136	1
137	0
138	0
139	0
140	0
141	0
142	0
143	0
144	0
145	0
146	0
147	0
148	0
149	0
150	0



BAZINGA!

Sheldonmir Cooperino

Se curtiu, deixa o like,
te inscreve no canal e
ativa as notificações.